ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Safe Guarding Mobile Phones against Phishing Attacks Using MobiFish

Soumya.M.S[1], Prof. Phani Ram Prasad[2]

PG student (M. Tech CNE), Bellary Institute of Technology and Management, Bellary, Karnataka, India[1]

Assistant Professor, Department of CSE. Bellary Institute of Technology and Management, Bellary, Karnataka, India[2]

ABSTRACT: Now a day the threat of phishing attacks on mobile computing platforms is getting increased as many of the users use mobile phones to access the net banking, Gmail, Face book and some other application. MobiFish is automated defense scheme, where users no need to make the final decision, but it is the users who finally remove the phishing app. Actually, they do not need to explicitly make the decision at all, since the only explanation for the login failure is a phishing attack. Here there is no need of developer to design the browser/app/website's UIs, MobiFish is compatible with all existing websites and apps. In this paper, we propose specialized form of phishing attacks which target at the persistent account registry function of mobile OSs. We employ the optical character recognition (OCR) technique to extract text from the screenshot of a login interface, which achieves better performance on mobile phones than on PCs.

KEYWORDS: Phishing attacks, OCR technique, mobile security, mobile computing.

I. INTRODUCTION

Phishing attack causes password leaking damage to mobile users. This attacks aims to get user private information such as login name, passwords, and credit card details by way of impersonating a legitimate entity. Even though security researchers have proposed many anti-phishing schemes, phishing attacks threat is not well mitigated, and we have a specialized form of phishing attacks that target persistent account registry function of mobile OSs. Several number of phishing sites revive and expire rapidly. Increase in phishing attacks on mobile computing platform is the main issue.

We need to design a defense scheme specifically to account registry function of mobile platform. We make us of optical character recognition (OCR) method to extract text from the screenshot of a login page, which attain better performance on mobile phones. So we will be able to find the claimed identity from the extracted text, and the real identity from the URL. We are implementing MobiFish on a Smartphone running the Android OS, which effectively detects and defend against mobile phishing attacks.

II. RELATED WORK

Web users have been suffering from phishing attacks since their first appearance in 2003. Researchers have proposed many solutions such as alert protection and phishing detection to defend against phishing attacks. Mobile phishing attacks could also be in the form of Emails or Short Messaging Services (SMS). Phishing emails usually request users to click a link to a fake website where the user is prompted to enter login credentials [1][6].

In addition, many phishing detection tools have been designed for phishing on PC web pages. Based on the methods used, they can be generally categorized into two groups: heuristics schemes and blacklist schemes. Heuristics schemes outperform blacklist schemes since they can deal with new phishing sites without having to wait for an update. Usually, heuristics schemes for phishing detection utilize other techniques such as machine learning techniques and search engine [8].

CANTINA [8] is a content based approach to detecting phishing websites, and it adopts TF-IDF information retrieval algorithms. The SMS phishing attacks (SMiShing) [2][3] usually trick users into visiting a fraudulent website or calling a phishing number, where the victims are enticed into providing the credentials. The fraudulent websites could be defended by WebFish. But the detection of the phishing voice calls is beyond the scope of this article. Most voice phishing (Vishing) uses the VoIP technique in which the phone number is dynamically generated, we left this part for future work.

Our previous work [4] proposed the WebFish and AppFish schemes. In this article, we present the new persistent account phishing attack which has been neglected by existing works. We resolve this vulnerability with the AccountFish scheme.

Marforio et al. [5] applied personalized security indicators (an image chosen by the user that is displayed in the login UI) to mobile apps. However, all these indicator-based approaches require the user to make the final decision. Felt et al. [7] proposed to add an always-present identity bar that displays the name of the current foreground application or the domain name of the current web page.

Bianchi et al. [9] implemented an identity indicator for apps in the system navigation bar, in which Extended Validation (EV) HTTPS infrastructure is used to validate the app developers.

III. PROPOSED SYSTEM

The block diagram of proposed system is shown below which includes Web Fish, App Fish and Account Fish. In the WebFish method URL will be taken as input, URL domain name verification, download HTML page, check for form presence, extract text from form and match against sensitive text and warning the user if that page contains any phishing link. In the AppFish during app installation App name verification, app login screen capture and verify sensitive texts are done, and during launching check outgoing SMS for sensitive text and check outgoing URL for domain will be done and warns the user, and in the AccountFish main process are if account name is null it is malicious app and account name is same as app name it is set as new app and added to the main menu and connection will be established.

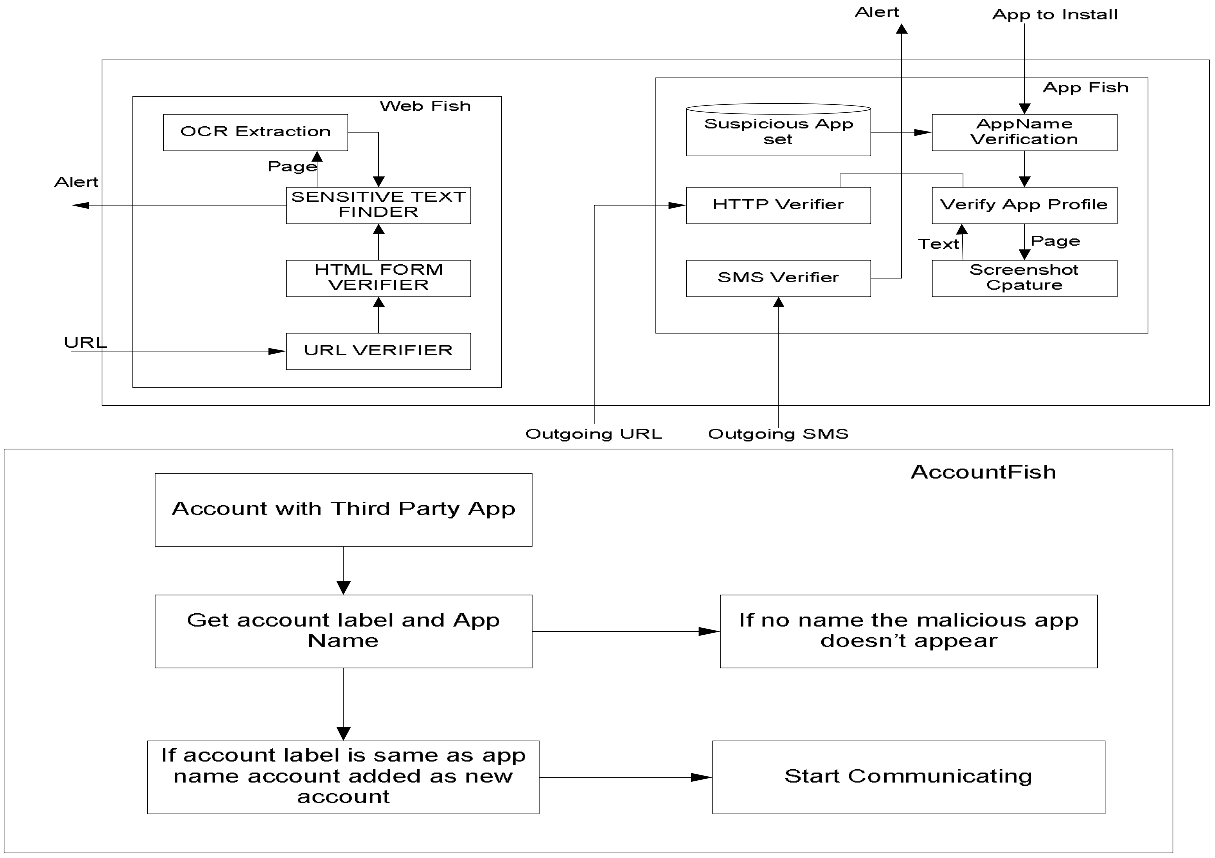

Fig 1: Block diagram for the proposed system

In this paper, we propose MobiFish technique for defending against mobile web pages, apps and persistent accounts. WebFish, AppFish and AccountFish includes different methods for recognizing malicious app, these includes different methods like Optical Character Recognition (OCR) to extract text from screenshot for checking URLs, second level domain name (SLD), suspicious App set (SAS) this contains untrusted apps and account mapping white list (AMWL) that contains all inconsistent legitimate apps.

IV. SYSTEM DESIGN

We present an automated lightweight scheme for mobile phishing defense named MobiFish. This consists of three major components named WebFish, AppFish, and AccountFish designed to protect mobile web pages, applications, and persistent accounts.

Work flow for WebFish

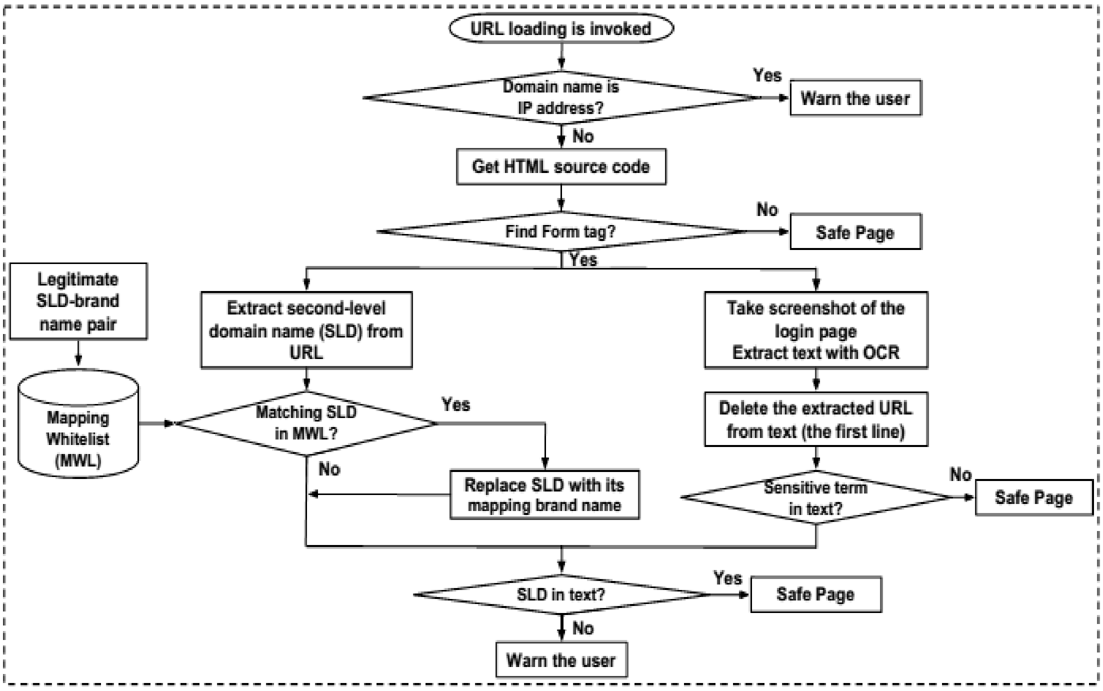
URL loading is invoked
Domain name is IP address?
Yes
Warn the user
No
Get HTML source code
Find Form tag?
No
Safe Page
Yes
Legitimate SLD-brand name pair
Extract second-level domain name (SLD) from URL
Take screenshot of the login page Extract text with OCR
Mapping Whitelist (MWL)
Matching SLD in MWL?
Yes
No
Replace SLD with its mapping brand name
Delete the extracted URL from text (the first line)
Sensitive term in text?
No
Safe Page
SLD in text?
Yes
Safe Page
No
Warn the user

Fig 2. work flow diagram for WebFish

This scheme starts with URL loading. It first scans URL to check domain name is an IP address, if domain name contains IP address it warns to user, If not it obtains the HTML source code of the loading page, and checks for the form tag. If form tag is found, it starts the extraction and verification of the identity, if form is not found then the page is safe. On one hand WebFish extracts SLD brand names as some of the branded enterprises uses the brand name as their second level domain name for their websites, the SLD from the URL that contains the actual identity of the site, and then the SLD extracted is indexed in Mapping White-List (MWL) in which it checks the name with legitimate SLD brand pair. If any match found with SLD-Brand name, then the original SLD is replaced if the SLD is not matched or not found then the site is considered as malicious and warns the user.

On the other hand, screenshot of the login page is taken and text from the screenshot is extracted with the Optical Character Recognition (OCR) technique, OCR is mechanical or electronic conversion of image to machine encoded text. Before checking sensitive terms it removes the first file from the text which may contain the phishing link and then it sends the sensitive terms to map extracted SLD with MWL. If it is not found, then site is marked as a phishing site and it warns the user and if SLD is found then legitimate brand name will be replace with existing. Design is based on the assumption that if the domain name of the phishing site appears in the fake login page of a legitimate entity, the user will check for the validity of the page.

Work Flow for AppFish

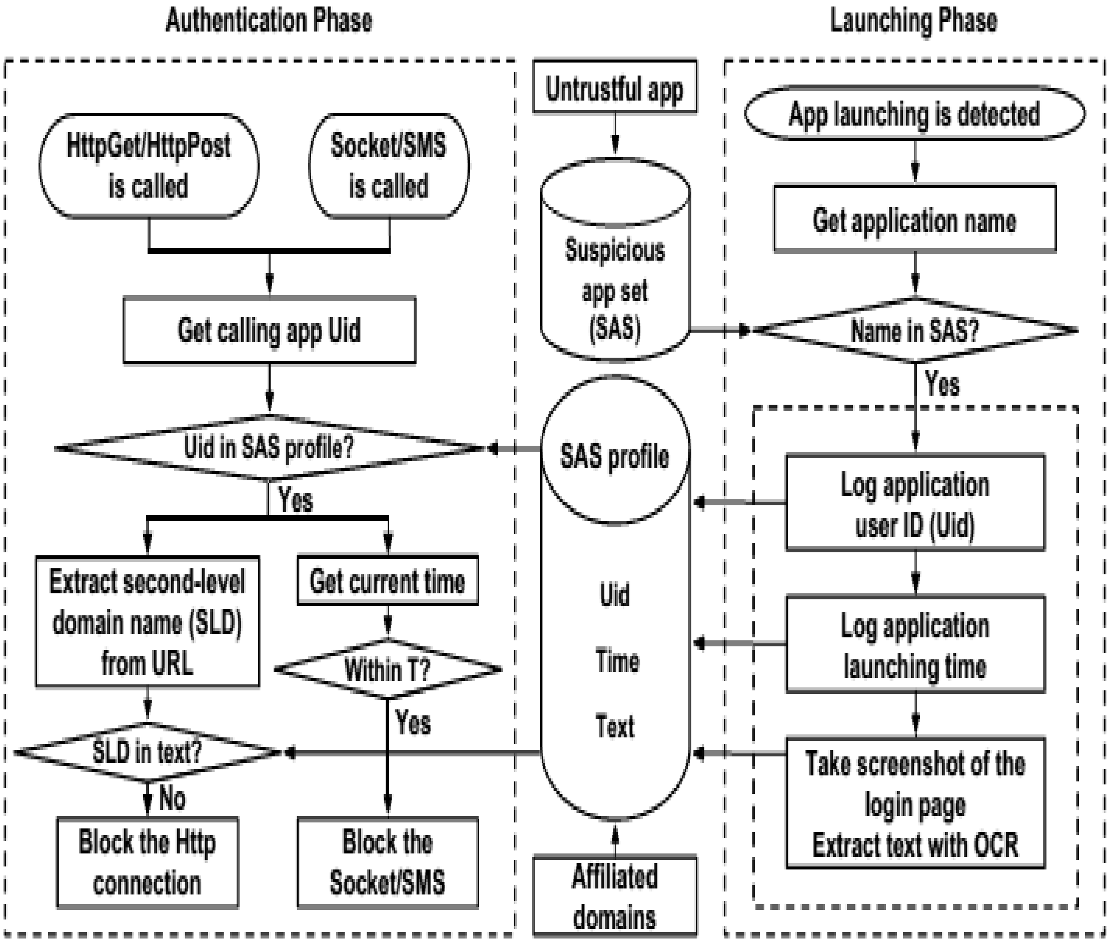

Fig 3. work flow diagram for AppFish

AppFish is designed to check the malicious apps present in the mobile, it maintains a database called Suspicious App Set (SAS) it contains user ID, launching time and screenshot. The app that are downloaded may be malicious and some of the malicious application can be identified while downloading app from the unauthorized site, this scheme works in two phases: launching phase and authentication phase, In launching phase, AppFish takes the name of each launching application and check for that name in SAS which contains all the untrusted app details. If it is found it takes a screenshot of the login page and extracts the text using OCR technique then the text along with application Uid, launching time of that app and profile details of the app.

After user enter the details and click submit the authentication phase starts legitimate application sends the user details to remote server and loads the data after identification are verified, the application loads data belonging to that account, if the application is malicious it will be unable to load the user data as it does not contain any user information, that are designed only to get user detail and ask re-entering information by showing user has entered wrong login id or password. The information may be requested through sms or through web notifications that needs user login, so Appfish before sending information it checks it is in SAS. If found HTTP connections are filtered till then other connections will be blocked for certain period of time T, by that time the user notices the malicious app and remove the app, and mean while AppFish ensures SLD name or domain name in SAS profile and notices it is malicious app. If app does not contain any such malicious activities then it will be installed and used.

The Work Flow of AccountFish

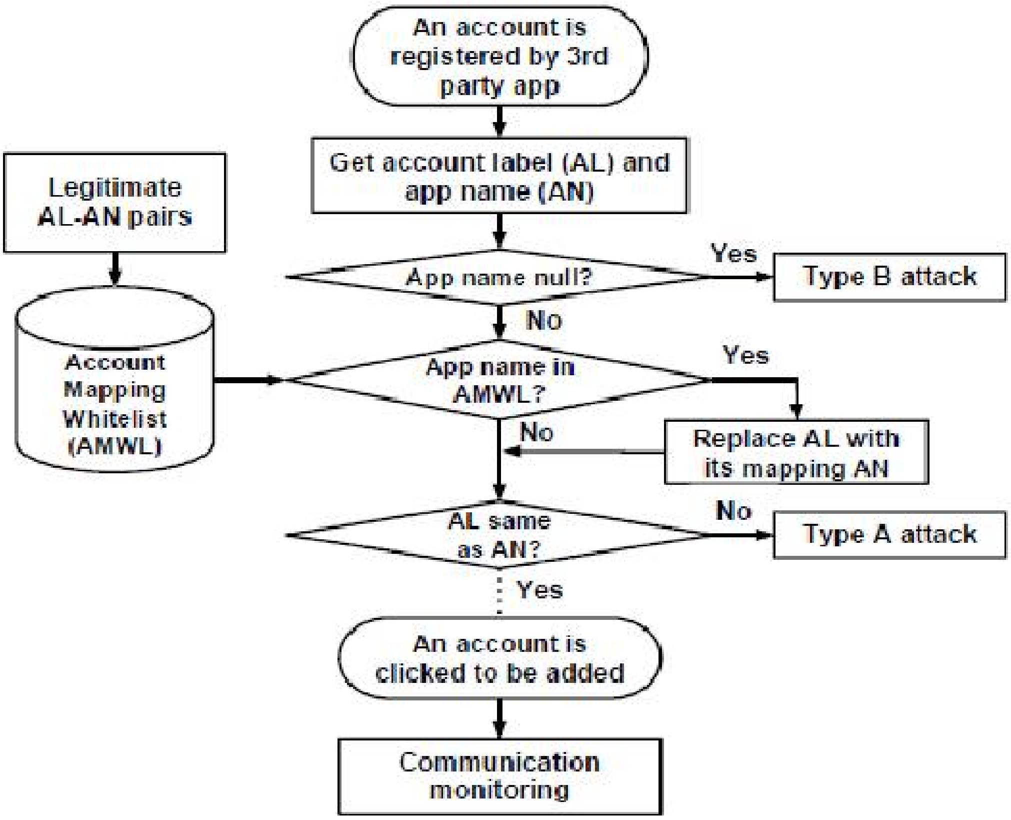
An account is registered by 3rd party app
Get account label (AL) and app name (AN)
Legitimate AL-AN pairs
App name null?
Yes
Type B attack
No
Account Mapping Whitelist (AMWL)
App name in AMWL?
Yes
Replace AL with its mapping AN
No
AL same as AN?
No
Type A attack
Yes
An account is clicked to be added
Communication monitoring

Fig 4. work flow diagram for AccountFish

AccountFish is to check account registry attacks that targets at persistent accounts, account registry attacks scan be of three types based on the identities the malicious app. In the type A attack, the malicious app appears to be a different app to the target account, which will be downloaded as one application and functions as another. In the type B attack, the malicious app does not appear in the main menu here the app will be not visible in the app list but you can see its details only in the storage or setting of the mobile. Detection mechanism for type A and B is to get the account label and application full name and compare the app name(AN) in main menu and account label(AL) in the account list. The type C attack, the malicious app shows up as the target app.

It should be able to check the account registration of during runtime that can be accomplished by modifying Android source code. If account label and account name are different then app may be malicious but some of legitimate app label are not same as resultant app names that problem is solved using account mapping white list (AMWL), which contains all suspicious app details where we check all legitimate AL with AN pairs. Mechanism use to detect C is similar to type C similar to AppFish the malicious activities cannot be found until the transformation of the information is done. Here we have bindToAuthenticator function that finds user action while adding account and monitor the process the app name will be used to check or filter the outgoing connections. Only the URLs with SLD will be allowed to communicate and suspicious activities sites are blocked for certain amount of time and user will be warned if it is malicious.

V. CONCLUSION

Recent years have witnessed the increasing threat of phishing attacks on mobile computing platforms. In this project we proposed MobiFish, a novel automated phishing defense scheme for mobile phones with android OS. In this work first we identified the weaknesses of the heuristics-based anti-phishing schemes that highly rely on the HTML

DOI: 10.15680/IJIRCCE.2016. 0405082

source code of web pages. MobiFish solve this issue by using OCR that accurately extracts text from the screenshot of the login page so that the claimed identity can be verified. Compared to existing anti-phishing schemes, MobiFish is lightweight as it works without using external search engines or machine learning techniques. MobiFish can also detect account phishing attacks. We are implementing MobiFish on Smartphone running the Android OS.

REFERENCES

[1]. J. Wang, T. Herath, R. Chen, A. Vishwanath, and H. Rao, “Phishing susceptibility: An investigation into the processing of a targeted spear phishing email,” IEEE Transactions on Professional Communication, vol. 55, no. 4, pp. 345–362, Dec 2012.

[2] A. Eshmawi and S. Nair, “Smartphone applications security: Survey of new vectors and solutions,” in In Proceedings of ACS International Conference on Computer Systems and Applications (AICCSA), pp.1-4, May 2013.

[3]A. Kang, J. Dong Lee, W. Kang, L. Barolli, and J. Park, “Security considerations for smart phone smishing attacks,” Advances in Computer Science and its Applications, vol. 279, pp. 467–473, 2014.

[4] L. Wu, X. Du, and J. Wu, “Mobifish: A lightweight anti-phishing scheme for mobile phones,” in Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN), pp.1-8, Aug 2014.

[5] C. Marforio, R. J. Masti, C. Soriente, K. Kostiainen, and S. Capkun,“Personalized security indicators to detect application phishing attacks in mobile platforms,” CoRR, vol. abs/1502.06824, 2015.

[6] A. Bergholz, J. De Beer, S. Glahn, M.-F. Moens, G. Paaß, and S. Strobel, “New filtering approaches for phishing email,” Journal of Computer Security, vol. 18, no. 1, pp. 7–35, Jan. 2010.

[7] A. P. Felt and D. Wagner, “Phishing on mobile devices,” In Proceedings of W2SP’11: WEB 2.0 Security and Privacy, pp.1-10, 2011.

[8] Y. Zhang, J. I. Hong, and L. F. Cranor, “Cantina: a content-based approach to detecting phishing web sites,” in Proceedings of the 16th international conference on World Wide Web (WWW), pp. 639-648, 2007.

[9] A. Bianchi, J. Corbetta, L. Invernizzi, Y. Fratantonio, C. Kruegel, and G. Vigna, “What the app is that? deception and countermeasures in the android user interface,” in Proceedings of the IEEE Symposium on Security and Privacy (SP),pp.931-948, May 2015.

[10] L. F. Cranor, S. Egelman, J. I. Hong, and Y. Zhang, “Phinding phish: Evaluating anti-phishing tools,” In Proceedings of The 14th Annual Network and Distributed System Security Symposium (NDSS), February, 2007.